Advanced Services for Internet Telephony

Jonathan Lennox

Thesis Proposal
Department of Computer Science
Columbia University

February 22, 2000

Overview

- Motivation
- Internet Telephony Architecture
- Related Work
- Service Creation Proposals
  - SIP CGI
  - Call Processing Language
- Feature Interaction Issues
- Research Plan

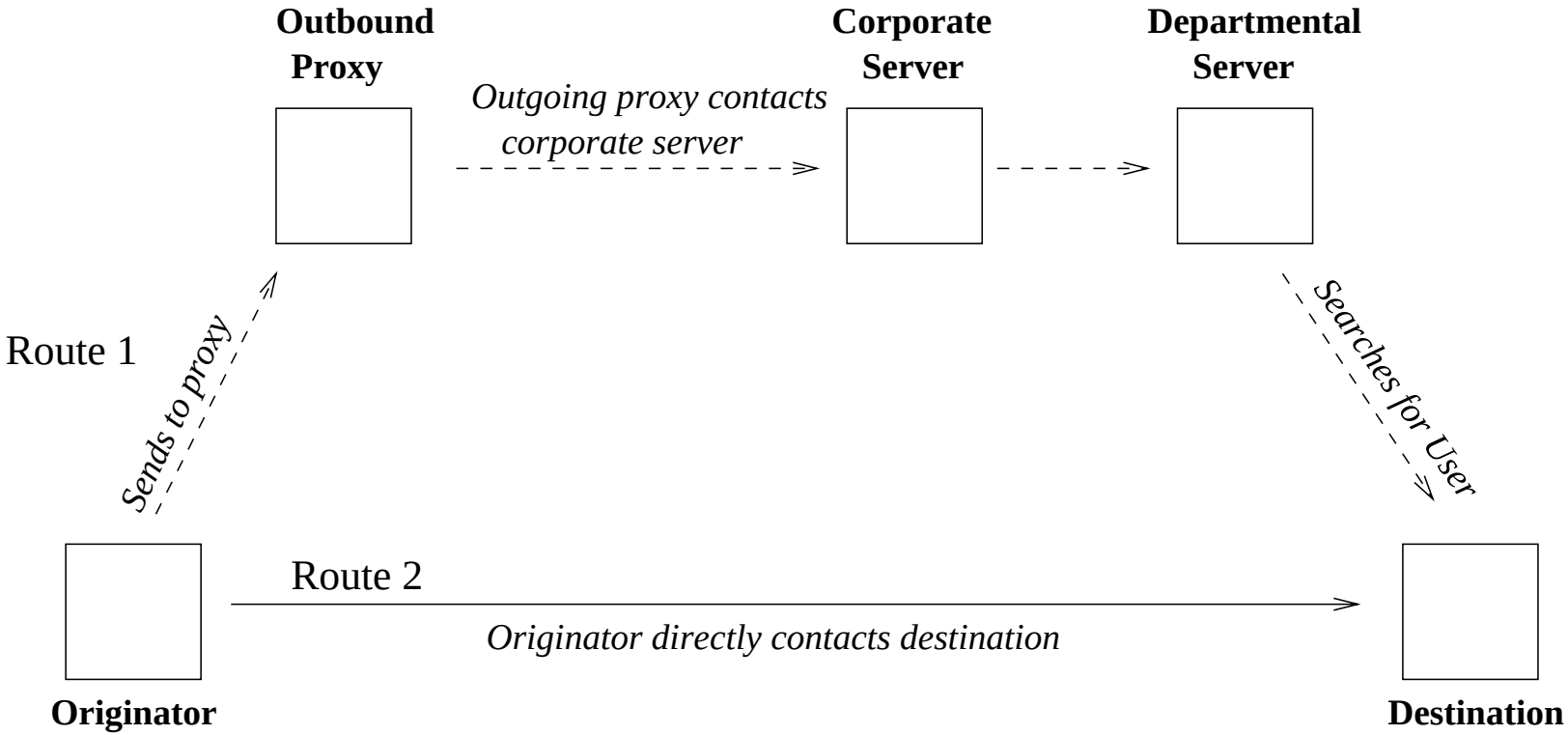
Motivation

- Internet telephony increasingly replacing circuit-switched network
- Need advanced telephony services for Internet telephony
  - Services comparable to traditional networks
    - Call forwarding, call blocking, time-of-day routing, ...
  - New or enhanced services
    - Interfaces with other parts of Internet
    - Elaborate, custom services
- Need new ways of creating such services

Internet Telephony Architecture

- End systems
  - Simple and complex telephones, PC telephony clients, automated systems
  - Originate and receive calls
  - Send media end-to-end
- Signalling servers
  - Proxy servers, gatekeepers, switches
  - Proxy, reject, or redirect calls
  - Locate users
  - Advanced services live here

Internet Telephony Architecture: Message Flow



- Originator may use a local server, or may communicate directly
- Messages may pass directly between end systems, or through signalling servers

Related Work

- Intelligent Network Services
  - Capability Sets
  - Service-Independent Building Blocks
- Mechanisms for Other Internet Applications
  - Web: CGI, Servlets
  - E-mail: Sieve
  - Lower Protocol Levels: Active Networks
- APIs
  - JAIN; Parlay; SIP Servlets
- Scripting Languages
  - PML; CPML

SIP CGI

- Characteristics:
  - Derived from HTTP CGI
  - Text-based API for low-level control of SIP servers
  - Intended for administrators and trusted users
- Benefits:
  - Language independent
  - All headers exposed
  - All parts of messages controllable
  - Components can be reused
  - Familiar environment
  - Easily extensible

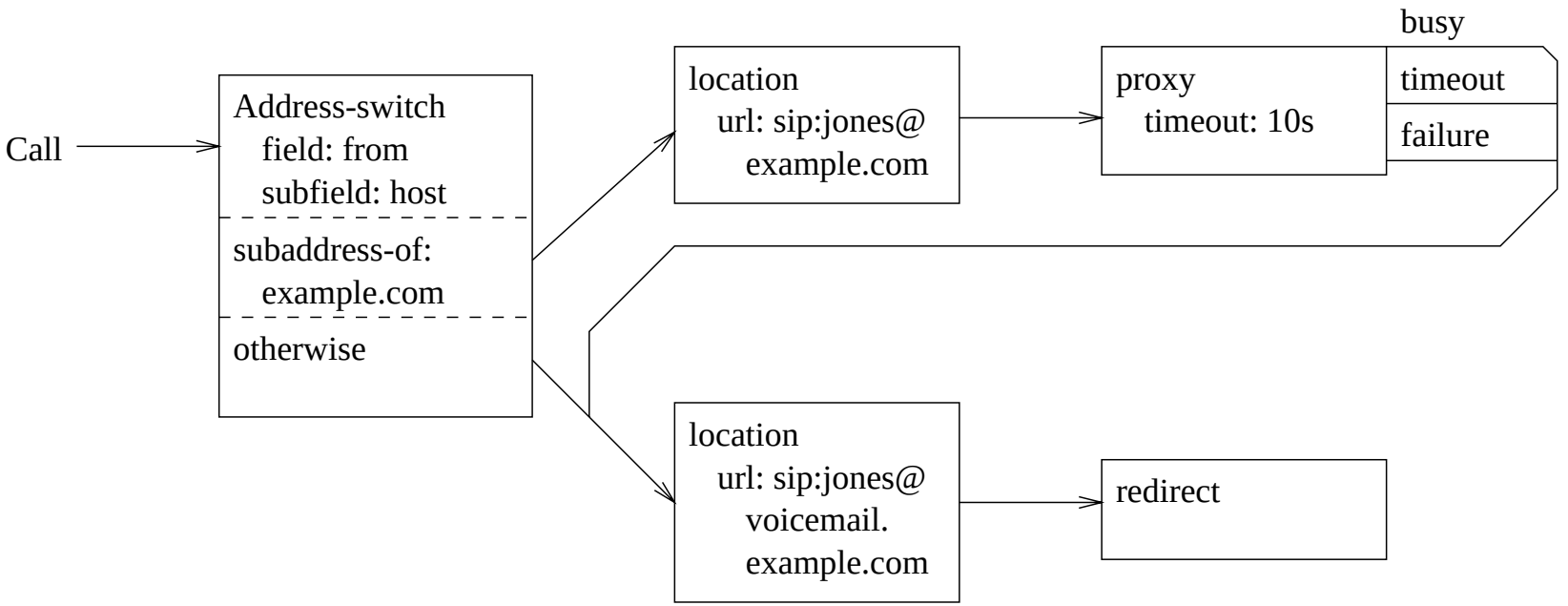
SIP CGI: Model

- Server invokes script when request or response arrives
- Server passes a SIP request through stdin and environment variables
- Script outputs actions through stdout: takes control of server actions
  - Send new response (e.g. failure status)
  - Proxy request
  - Forward response
- Script can maintain its own state with server
  - State tokens
  - Whether to re-execute script on subsequent messages in the transaction

Call Processing Language

- Creatable and editable both by simple graphical tools and by humans
- Independent of underlying signalling protocol
- Safe to run in servers
  - Automatically verifiable when uploaded to server
  - Resource usage (memory, processing time) inherently limited
  - Not Turing-complete: no loops, variables, recursion
  - Can’t interact with inappropriate parts of the server
  - Predictable behavior

CPL: Example



Graphical structure represented textually using XML

CPL: Structure

Abstract Structure	**XML**
Nodes and outputs — “boxes” and “arrows”	Nested Tags
Nodes have parameters	Tag attributes
Start from single root “call” node	Top-level `call` tag
Acyclic graph	XML document is a tree; `link` connects branches

- Progress down acyclic graph
- Follow one output of each node, based on outcome
- Continue until we get to a node with no outputs

Feature Interaction in Internet Telephony

- Feature Interaction: several features or services, operating simultaneously, interact so as to interfere with the desired operation of some of the features
- Internet Telephony resolves some sorts of F.I., but adds new problems
  - Distributed nature of the Internet
  - Features created by amateur feature designers
  - Media packets travel end-to-end
  - End systems have control of call state
  - Lack of address scarcity
  - Trust model
  - New features:
    - Forking Proxies
    - Request Expiration

Internet Telephony Feature Interaction: Examples

Request Forking and Call Forward to Voicemail A call is forked to two locations, one of which is a voicemail server; the voicemail always answers before the human can pick up.

Multiple Expiration Timers Two signalling servers both have special timeout behavior programmed, and their timeouts are based on the expiration time of the request. Either one may be executed first.

Outgoing Call Screening and Call Forwarding An administrator wants to block calls to a particular address; outside of the domain, calls are forwarded from a non-blocked address to the blocked one.

Outgoing Call Screening and End-to-end Connectivity An administrator wants to block calls to a particular address; the end system simply does not use the administrator's outgoing proxy.

Incoming Call Screening and Polymorphic Identity A user wants to block calls from certain callers; the offending callers alter the identity in the call request to a non-blocked address.

Internet Telephony Feature Interaction: Possible Solutions

- Explicitness
- Universal authentication
- Network-level administrative restrictions
- Verification testing

Research Plan: Service Creation Models

- SIP CGI
  - Essentially complete
  - Submit as Informational RFC once HTTP CGI is published
- CPL
  - Core proposal
    - New Internet-Draft to be submitted
    - Work item of IETF IPTel Working Group
  - Extensions
    - In-call or end-of-call actions
    - Other devices: end systems, voicemail
    - Other Internet communication: instant messages, presence, e-mail
    - Administrative scripts
    - Fine-grained media knowledge

Research Plan: Implementations, Performance, F.I.

- Implementations
  - SIP CGI implemented in Columbia SIPD
  - CPL will be implemented in it
- Performance
  - Determine how to evaluate performance of SIP servers, services
  - Benchmark our server and these service creation models
- Feature Interaction
  - Evaluate proposed solutions
  - Implement SIP Caller Preferences
  - Construct CPL simulator for verification

Plan for Completion of Research

- Short term:
  - SIP CGI will be published as an RFC as soon as status of HTTP-CGI is resolved
  - Finish CPL, to move towards Proposed Standard
  - Implement CPL in Columbia SIPD
- Longer term:
  - Investigate CPL extensions
  - Implement feature interaction proposals
  - Investigate performance
- Finish by September 2001

Backup Slides

Example SIP CGI Script

```
#! /usr/bin/perl

if ($ENV{SIP_FROM} =~ "sip:abc@") {
    print "SIP/2.0 600 I can't talk right now\n\n";
}
```

Example CPL Script

```
<?xml version="1.0" ?>
<!DOCTYPE call SYSTEM "cpl.dtd">

<call>
  <address-switch field="origin" subfield="host">
    <address subaddress-of="example.com">
      <location url="sip:jones@example.com">
        <proxy>
          <busy> <link ref="voicemail" /> </busy>
          <noanswer> <link ref="voicemail" /> </noanswer>
          <failure> <link ref="voicemail" /> </failure>
        </proxy>
      </location>
    </string>
    <otherwise>
      <location url="sip:jones@voicemail.example.com"
                id="voicemail">
        <redirect />
      </location>
    </otherwise>
  </string-switch>
</call>
```